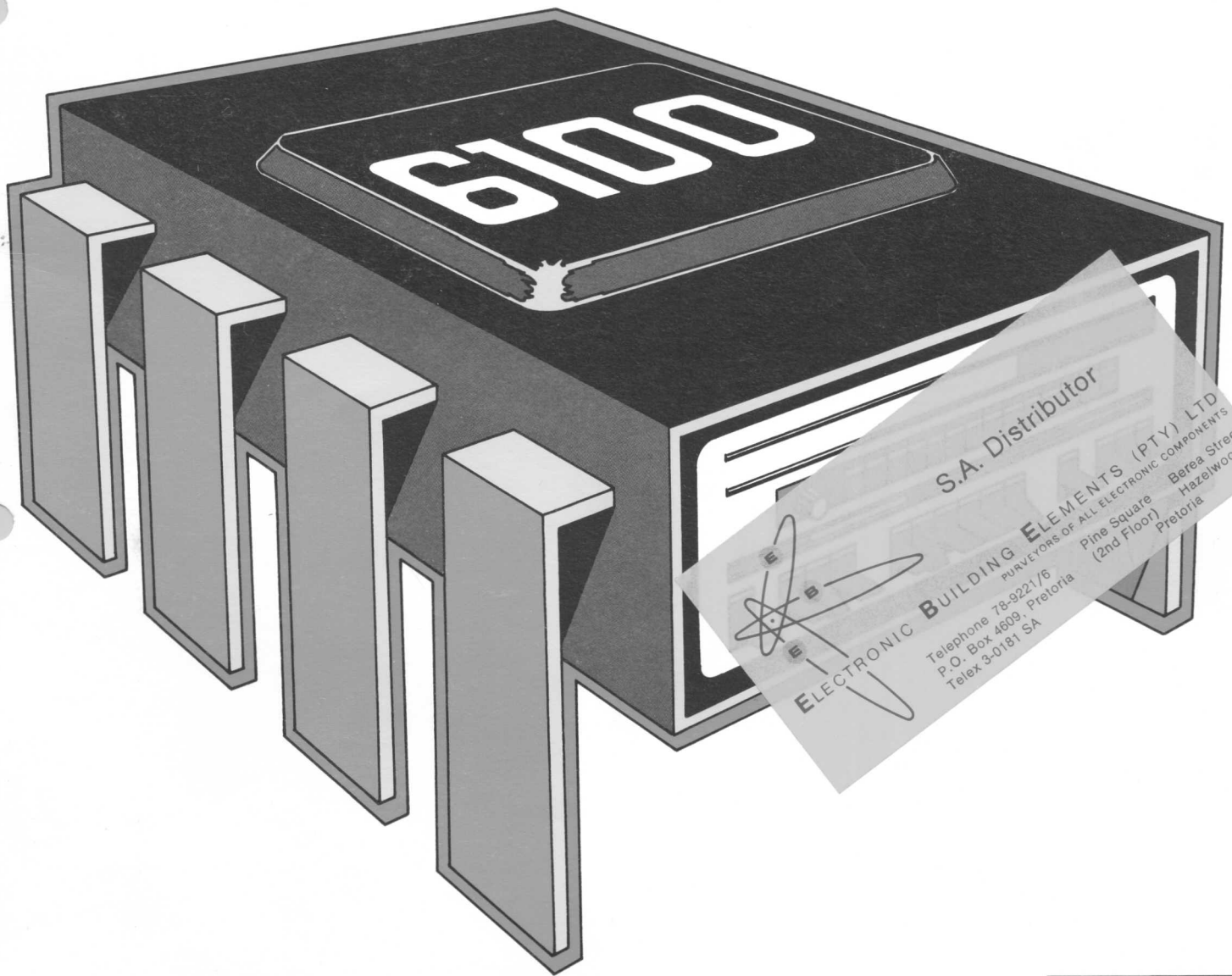


# INTERSIL IM6100 CMOS 12 BIT MICROPROCESSOR



**INTERSIL**  
**CMOS/LSI**

# CONTENTS

1

## SECTIONS

<b>Introduction</b>	<b>2</b>
IM6100 Microprocessor	2
Features	3
Design Features	3
Interface Features	3
Architectural Features	3
Applications	3
<b>Pin Assignments</b>	<b>4</b>
<b>Specifications</b>	<b>5</b>
Absolute Maximum Ratings	5
DC Characteristics	5
Package Dimensions	5
Ordering Informations	5
<b>Architecture</b>	<b>6</b>
Accumulator (AC)	6
Link (L)	6
MQ Register (MQ)	6
Memory Address Register (MAR)	6
Program Counter (PC)	6
Arithmetic and Logical Unit (ALU)	6
Temporary Register (TEMP)	6
Instruction Register (IR)	6
Multiplexer (DX)	6
Major State Generator and the Programmed Logic Array	6
PLA Output Latch	6
Memory and Device Control,	
ALU and Reg Transfer Logic	6
Timing and State Control	7
AC Characteristics	7
<b>Memory and Processor Instructions</b>	<b>8</b>
Memory Organization	8
Memory Reference Instructions (MRI)	8
<b>Operate Instructions</b>	<b>10</b>
Group 1 Microinstructions	11
Group 2 Microinstructions	12
Group 3 Microinstructions	13
<b>Input/Output Transfer Instructions (IOT)</b>	<b>14</b>
IOT Instruction Format	14
Programmed Data Transfer	14
Interrupt Transfer	15
Programmed Interrupt Transfers	15
Device Interrupt Grant Timing	15
Control Panel Interrupt Transfers	16
Direct Memory Access (DMA)	18
<b>Internal Priority Structure</b>	<b>19</b>
Ifetch	20
Reset	20
Run/Halt	20
<b>PDP-8/E* Compatibility</b>	<b>21</b>
<b>Applications</b>	<b>22</b>
All CMOS System	22
General Purpose IM6100 System	22
256 × 12 RAM, 2K × 12 P/ROM Memory System	23
Transparent Control Panel	23
IM6100 to CMOS RAM Interface	24

## FIGURES

FIGURE 1	IM6100 Functional Block Diagram	6
FIGURE 2	IM6100 Timing and State Signals	7
FIGURE 3	Memory Reference Instruction (MRI) Format	8
FIGURE 4	Basic OPR Instruction Format	10
FIGURE 5	Group 1 Microinstruction Format	11
FIGURE 6	Group 2 Microinstruction Format	12
FIGURE 7	Group 3 Microinstruction Format	13
FIGURE 8	OSR Instruction Timing	13
FIGURE 9	IOT Instruction Format/States/Execution Time	14
FIGURE 10	Input-Output Instruction Timing	15
FIGURE 11	Device Interrupt Grant Timing	15
FIGURE 12	Device Interrupt Grant Reset Timing	15
FIGURE 13	Interrupt Enable FF On (ION) Timing	16
FIGURE 14	Control Panel Interrupt Grant Timing	16
FIGURE 15	"DCA INDIRECT" in Control Panel Routine	17
FIGURE 16	"ION; JMP 0000 <sub>8</sub> " in Control Panel Routine	16
FIGURE 17	Direct Memory Access (DMA)	18
FIGURE 18	Major Processor States and Number of Clock Cycles in Each State	19
FIGURE 19	Reset Timing	20
FIGURE 20	Run/Halt Timing	20
FIGURE 21	"Single Stepping" with RUN/HLT	20
FIGURE 22	PDP-8/E Compatibility	21
FIGURE 23	CMOS System	22
FIGURE 24	General Purpose IM6100 System	22
FIGURE 25	256 × 12 RAM-2K × 12 P/ROM System	23
FIGURE 26	Transparent Control Panel	23
FIGURE 27	CMOS RAM Interface	24

## TABLES

TABLE 1	Memory Reference Instructions	9
TABLE 2	Group 1 Operate Microinstructions	11
TABLE 3	Group 2 Operate Microinstructions	12
TABLE 4	Group 3 Operate Microinstructions	13
TABLE 5	Programmed I/O Control Lines	14
TABLE 6	Processor IOT Instructions	16

# INTRODUCTION

---

2

## IM6100 MICROPROCESSOR

Since its founding on July 26, 1967, INTERSIL INC. has offered its customers advanced products utilizing the semiconductor industry's most technologically sophisticated processes for the manufacture of practical, economical devices.

The Silicon Gate CMOS process, which was developed at Intersil in 1972, offers a semiconductor structure resulting in packing densities which surpass the conventional metal gate process 3:1. Additionally, circuit performance is improved 2:1.

Mass production experience with the Silicon Gate CMOS process, through previously announced 256 and 1024 bit CMOS RAMs, has lead to the practicality of introducing the IM6100 microprocessor.

The IM6100 is a single address, fixed word length, parallel transfer microprocessor using 12-bit, two's complement arithmetic. The processors recognize the instruction set of Digital Equipment Corporation's PDP8/E minicomputer. The internal circuitry is completely static and is designed to operate at any speed between DC and the maximum operating frequency. Two pins are available to allow for an external crystal thereby eliminating the need for clock generators and level translators. The crystal can be removed and the process clocked by an external clock generator. A 12-bit memory accumulator ADD instruction, using a +5 volt supply, is performed in 5 $\mu$ sec by the IM6100, in 6 $\mu$ sec by the IM6100C and in 2.5 $\mu$ sec by the IM6100A using a +10 volt supply. The device design is optimized to minimize the number of external components required for interfacing with standard memory and peripheral devices.

## FEATURES

### DESIGN

- Silicon Gate Complementary MOS
- Fully Static-0 to 8 MHz
- Single Power Supply
  - IM6100/C  $V_{cc}=5$  volts
  - IM6100A  $V_{cc}=10$  volts
- Crystal Controlled On Chip Timing
- Low Power Dissipation  $< 10$  mW @ 4 MHz @ 5 volts
- Single Power Supply  $4V \leq V_{cc} \leq 11V$
- TTL Compatible at 5 Volts
- Excellent Noise Immunity
- $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  Operation

### INTERFACE

- Memory—Any Speed
- Control Panel
- Switch Register
- Asynchronous CPU—Memory and CPU—Device Communication
- 64 I/O Devices with PDP-8/E Compatible Interface
- Device Controlled Input-Output
- All Control Signals Produced By The CPU
- Power-on Initialize

### ARCHITECTURAL

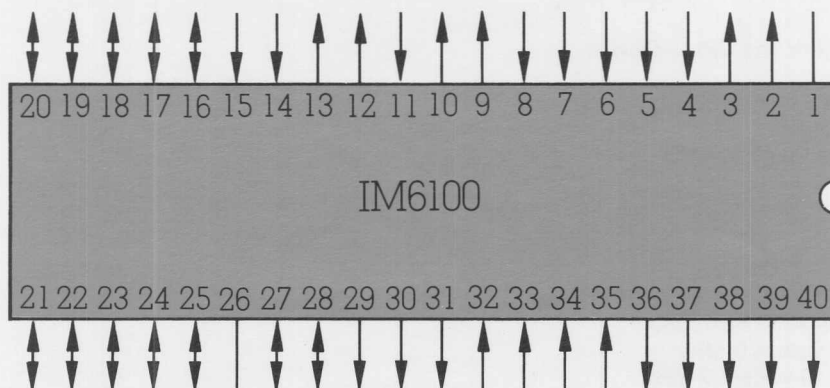
- Executes PDP-8/E, Instruction Set
- Direct, Indirect, and Autoindexed Memory Addressing
- 12-Bit Memory Accumulator ADD Instruction
  - IM6100  $5\mu\text{sec}$  @  $+5$  volts/4.0 MHz
  - IM6100A  $2.5\mu\text{sec}$  @  $+10$  volts/8.0 MHz
  - IM6100C  $6\mu\text{sec}$  @  $+5$  volts/3.3 MHz
- Input-Output Instruction
  - IM6100  $8.5\mu\text{sec}$  @  $+5$  volts/4.0 MHz
  - IM6100A  $4.25\mu\text{sec}$  @  $+10$  volts/8 MHz
  - IM6100C  $10.2\mu\text{sec}$  @  $+5$  volts/3.3 MHz
- Single-Clock, Single-Instruction Capability
- Direct Memory Access (DMA)
- Interrupt
- Dedicated Control Panel Features

## APPLICATIONS

- Intelligent Computer Terminals
- POS Terminals
- Portable Terminals
- Aerospace/Satellite System
- Automotive Systems
- Remote Data Acquisition Systems
- Process Control
- Instrumentation
- Medical Electronics
- Displays
- Traffic Control
- Navigation

PIN	SYMBOL	ACTIVE LEVEL	DESCRIPTION
1	V <sub>cc</sub>		Supply voltage.
2	RUN	H	The signal indicates the runstate of the CPU and may be used to power down the external circuitry
3	DMAGNT	H	Direct Memory Access Grant—DX lines are three-state.
4	DMAREQ	L	Direct Memory Access Request—DMA is granted at the end of the current instruction. Upon DMA grant, the CPU suspends program execution until the DMAREQ line is released.
5	CPREQ	L	Control Panel Request—a dedicated interrupt which bypasses the normal device interrupt request structure.
6	RUN/HLT	L	Pulsing the Run/Halt line causes the CPU to alternately run and halt by changing the state of the internal RUN/HLT flip flop.
7	RESET	L	Clears the AC and loads 7777 <sub>h</sub> into the PC. CPU is halted.
8	INTREQ	L	Peripheral device interrupt request.
9	XT <sub>A</sub>	H	External coded minor cycle timing—signifies input transfers to the IM6100.

PIN	SYMBOL	ACTIVE LEVEL	DESCRIPTION
10	LXMAR	H	The Load External Address Register is used to store memory and peripheral address externally.
11	WAIT	L	Indicates that peripherals or external memory is not ready to transfer data. The CPU state gets extended as long as WAIT is active. The CPU is in the lowest power state with clocks running.
12	XT <sub>B</sub>	H	External coded minor cycle timing—signifies output transfers from the IM6100.
13	XT <sub>C</sub>	H	External coded minor cycle timing—used in conjunction with the Select Lines to specify read or write operations.
14	OSC OUT		Crystal input to generate the internal timing (also external clock input).
15	OSC IN		See Pin 14—OSC OUT (also external clock ground)
16	DX <sub>0</sub>		DataX—multiplexed data in, data out and address lines.
17	DX <sub>1</sub>		See Pin 16—DX <sub>0</sub> .
18	DX <sub>2</sub>		See Pin 16—DX <sub>0</sub> .
19	DX <sub>3</sub>		See Pin 16—DX <sub>0</sub> .
20	DX <sub>4</sub>		See Pin 16—DX <sub>0</sub> .



PIN	SYMBOL	ACTIVE LEVEL	DESCRIPTION
21	DX <sub>5</sub>		See Pin 16—DX <sub>0</sub> .
22	DX <sub>6</sub>		See Pin 16—DX <sub>0</sub> .
23	DX <sub>7</sub>		See Pin 16—DX <sub>0</sub> .
24	DX <sub>8</sub>		See Pin 16—DX <sub>0</sub> .
25	DX <sub>9</sub>		See Pin 16—DX <sub>0</sub> .
26	GND		Ground
27	DX <sub>10</sub>		See Pin 16—DX <sub>0</sub> .
28	DX <sub>11</sub>		See Pin 16—DX <sub>0</sub> .
29	LINK	H	Link flip flop.
30	DEVSEL	L	Device Select for I/O transfers.
31	SWSEL	L	Switch Register Select for the OR THE SWITCH REGISTER INSTRUCTION (OSR). OSR is a Group 2 Operate Instruction which reads a 12 bit external switch register and OR's it with the contents of the AC.
32	C <sub>0</sub>	L	Control line inputs from the peripheral device during an I/O transfer (Table 5).

PIN	SYMBOL	ACTIVE LEVEL	DESCRIPTION
33	C <sub>1</sub>	L	See Pin 32—C <sub>0</sub> .
34	C <sub>2</sub>	L	See Pin 32—C <sub>0</sub> .
35	SKP	L	Skips the next sequential instruction if active during an I/O instruction.
36	IFETCH	H	Instruction Fetch Cycle
37	MEMSEL	L	Memory Select for memory transfers.
38	CPSEL	L	The Control Panel Memory Select becomes active, instead of the MEMSEL, for control panel routines. Signal may be used to distinguish between control panel and main memories.
39	INTGNT	H	Peripheral device Interrupt Grant
40	DATAF	H	Data Field pin indicates the execute phase of indirectly addressed AND, TAD, ISZ and DCA instructions so that the data transfers are controlled by the Data Field, DF, and not the Instruction Field, IF, if Extended Memory Control hardware is used to extend the addressing space from 4K to 32K words.

# SPECIFICATIONS

5

## ABSOLUTE MAXIMUM RATINGS

Supply Voltage

IM6100/C +4.0V to +7.0V

Operating Temperature Range

IM6100A +4.0V to 11.0V

Commercial

0°C to +75°C

Input or Output Voltage Applied

GND -0.3V to  $V_{CC} + 0.3V$

Industrial

-40°C to +85°C

Storage Temperature Range

-65°C to +125°C

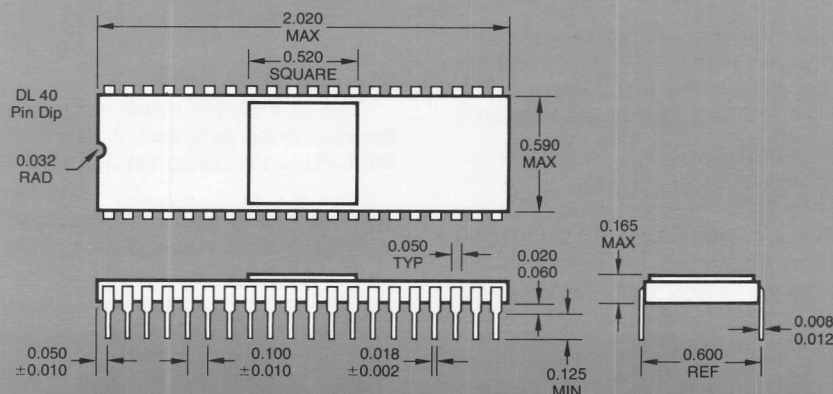
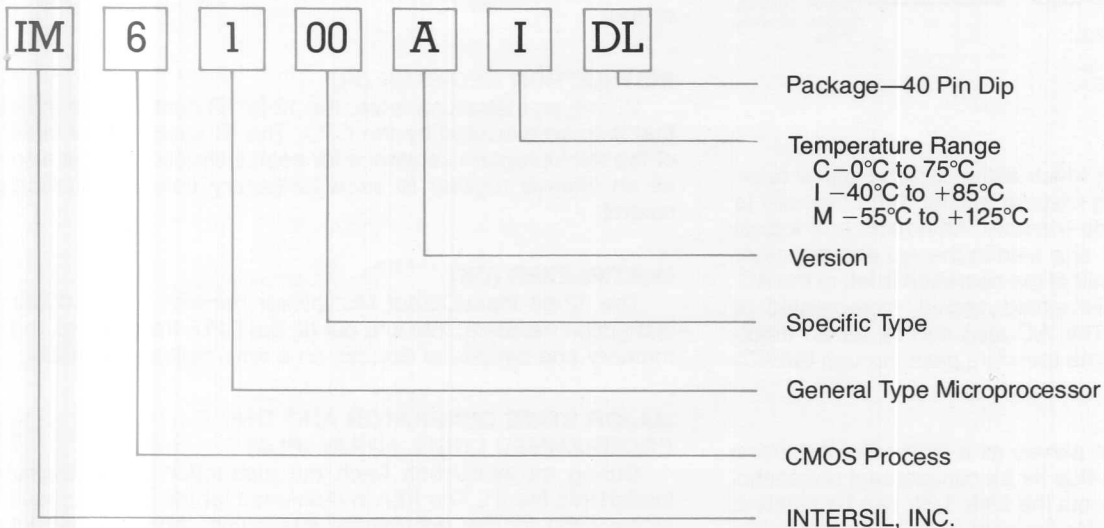
Military

-55°C to +125°C

DC CHARACTERISTICS  $V_{CC} = 5.0V \pm 10\%$  (IM6100),  $10.0V \pm 10\%$  (IM6100A),  $T_A$  = Commercial, Industrial or Military

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Logical "1" Input Voltage	$V_{IH}$		70% $V_{CC}$			V
Logical "0" Input Voltage	$V_{IL}$				20% $V_{CC}$	V
Input Leakage	$I_{IL}$	$0V \leq V_{IN} \leq V_{CC}$	-1.0		1.0	$\mu A$
Logical "1" Output Voltage	$V_{OH2}$	$I_{OUT} = 0$	$V_{CC} - 0.01$			V
Logical "1" Output Voltage	$V_{OH1}$	$I_{OH} = -0.2mA$	2.4			V
Logical "0" Output Voltage	$V_{OL2}$	$I_{OUT} = 0$			GND + 0.01	V
Logical "0" Output Voltage	$V_{OL1}$	$I_{OL} = 1.6 mA$			0.45	V
Output Leakage	$I_O$	$0V \leq V_o \leq V_{CC}$	-1.0		1.0	$\mu A$
Supply Current	$I_{CC}$	$V_{CC} = 5.0$ volts $V_{CC} = 10.0$ volts $C_L = 50 pF$ ; $T_A = 25^\circ C$ $F_{CLOCK} =$ Operating Frequency			2.5 10.0	$mA$ $mA$
Input Capacitance	$C_{IN}$			5.0		pF
Output Capacitance	$C_O$			8.0		pF

## ORDERING INFORMATION Circuit marking and product code explanation



## 6

FIGURE 1



**LINK (L)**

MQ REGISTER (MQ)

### MEMORY ADDRESS REGISTER (MAR)

**PROGRAM COUNTER (PC)**

## ARITHMETIC AND LOGICAL UNIT (ALU)

### TEMPORARY REGISTER (TEMP)

### INSTRUCTION REGISTER (IR)

### MULTIPLEXER (DX)

## MAJOR STATE GENERATOR AND THE PROGRAMMED LOGIC ARRAY (PLA)

### PLA OUTPUT LATCH

## MEMORY AND DEVICE CONTROL, ALU AND REG TRANSFER LOGIC

The Memory and Device Control Unit provides external control signals to communicate with peripheral devices (DEVSEL), switch register (SWSEL), memory (MEMSEL) and/or control panel memory (CPSEL). During I/O instructions this unit also modifies the PLA outputs depending on the states of the four device control lines (SKP, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>). The ALU and Register Transfer Logic provides the control signals for the internal register transfers and ALU operation.

### TIMING AND STATE CONTROL

The IM6100 generates all the timing and state signals internally. A crystal is used to control the CPU operating frequency. The CPU divides the crystal frequency by two. With a 4MHz crystal, the internal states will be of 500ns duration. The major timing states are described in Figure 2.

**T<sub>1</sub>** For memory reference instructions, a 12-bit address is sent on the DataX, DX, lines. The Load External Address Register, LXMAR, is used to clock an external register to store the address information externally, if required. When executing an Input-Output I/O instruction, the instruction being executed is sent on the DX lines to be stored externally. The external address register then contains the device address and control information.

Various CPU request lines are priority sampled if the next cycle is an Instruction Fetch cycle. Current state of the CPU is available externally.

**T<sub>2</sub>**

Memory/Peripheral data is read for an input transfer (READ). WAIT controls the transfer duration. If WAIT is active during input transfers, the CPU waits in the T<sub>2</sub> state. The wait duration is an integral multiple of the crystal frequency—250ns for 4MHz.

For memory reference instructions, the Memory Select, MEMSEL, line is active. For I/O instructions the Device Select, DEVSEL, line is active. Control lines, therefore, distinguish the contents of the external register as memory or device address.

External device sense lines, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, and SKP, are sampled if the instruction being executed is an I/O instruction.

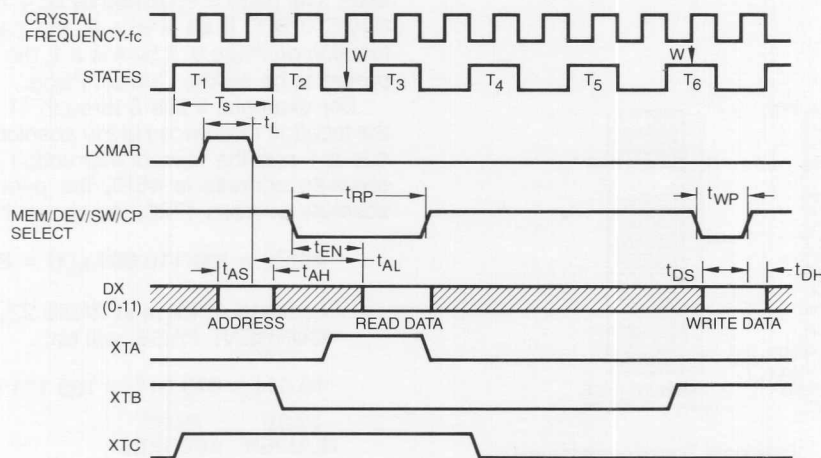
Control Panel Memory Select, CPSEL, and Switch Register Select, SWSEL, become active low for data transfers between the IM6100 and Control Panel Memory and the Switch Register, respectively.

**T<sub>3</sub>, T<sub>4</sub>, T<sub>5</sub>** ALU operation and internal register transfers.

**T<sub>6</sub>**

This state is entered for an output transfer (WRITE). The address is defined during T<sub>1</sub>. WAIT controls the time for which the Write data must be maintained.

## FIGURE 2



IM6100 TIMING AND STATE SIGNALS

### AC CHARACTERISTICS (T<sub>A</sub> = 25°C), Derate 0.390/°C

PARAMETER	SYMBOL	IM6100 V <sub>CC</sub> = 5.0 f <sub>c</sub> = 4MHz	IM6100A V <sub>CC</sub> = 10.0 f <sub>c</sub> = 8 MHz	IM6100C V <sub>CC</sub> = 5.0 f <sub>c</sub> = 3.3MHz	UNITS
Major State Time	T <sub>S</sub>	500	250	600	ns
LXMAR Pulse Width	t <sub>L</sub>	240	120	280	ns
Address Setup Time	t <sub>AS</sub>	50	30	80	ns
Address Hold Time	t <sub>AH</sub>	250	125	280	ns
Access Time From LXMAR	t <sub>AL</sub>	500	250	600	ns
Output Enable Time	t <sub>EN</sub>	240	120	280	ns
Read Pulse Width	t <sub>RP</sub>	700	350	800	ns
Write Pulse Width	t <sub>WP</sub>	240	120	280	ns
Data Setup Time	t <sub>DS</sub>	240	120	280	ns
Data Hold Time	t <sub>DH</sub>	100	50	160	ns

The IM6100 instructions are 12-bit words stored in memory. The IM6100 makes no distinction between instructions and data; it can manipulate instructions as stored variables or execute data as instructions when it is programmed to do so. There are three general classes of IM6100 instructions. They are referred to as Memory Reference Instruction (MRI), Operate Instruction (OPR) and Input/Output Transfer Instruction (IOT).

Before proceeding further, we will discuss the Specific Memory Organization with which the IM6100 interfaces.

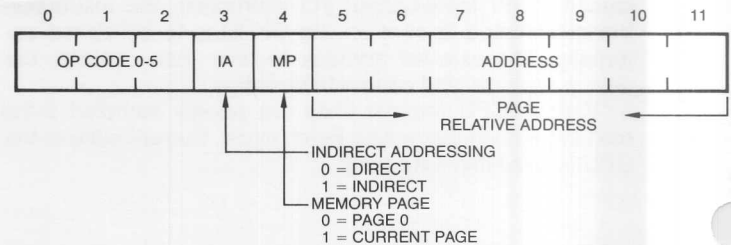
### MEMORY ORGANIZATION

The IM6100 has a basic addressing capacity of 4096 12-bit words. The addressing capacity may be extended by Extended Memory Control hardware. The memory system is organized in 4096 word blocks, called MEMORY FIELDS. The first 4096 words of memory are in Field 0. If a full 32K of memory is installed, the uppermost Memory Field will be numbered 7. In any given Memory Field every location has a unique 4 digit octal (12 bit binary) address, 0000<sub>8</sub> to 7777<sub>8</sub> (0000<sub>10</sub> to 4095<sub>10</sub>). Each Memory Field is subdivided into 32 PAGES of 128 words each. Memory Pages are numbered sequentially from Page 00<sub>8</sub>, containing addresses 0000-0177<sub>8</sub>, to Page 37<sub>8</sub>, containing addresses 7600<sub>8</sub>-7777<sub>8</sub>. The first 5 bits of a 12-bit MEMORY ADDRESS denote the PAGE NUMBER and the low order 7 bits specify the PAGE ADDRESS of the memory location within the given Page.

### MEMORY REFERENCE INSTRUCTIONS (MRI)

The Memory Reference Instructions operate on the contents of a memory location or use the contents of a memory location to operate on the AC or the PC. The first 3 bits of a Memory Reference Instruction specify the operation code, or OPCODE, and the low order 9 bits, the OPERAND address, as shown in Figure 3.

FIGURE 3



### MEMORY REFERENCE INSTRUCTION FORMAT

Bits 5 through 11, the PAGE ADDRESS, identify the location of the OPERAND on a given page, but they do not identify the page itself. The page is specified by bit 4, called the CURRENT PAGE OR PAGE 0 BIT. If bit 4 is a 0, the page address is interpreted as a location on Page 0. If bit 4 is a 1, the page address specified is interpreted to be on the Current Page.

For example, if bits 5 through 11 represent 123<sub>8</sub> and bit 4 is a 0, the location referenced is the absolute address 0123<sub>8</sub>. However, if bit 4 is a 1 and the current instruction is in a memory location whose absolute address is 4610<sub>8</sub>, the page address 123<sub>8</sub> designates the absolute address 4723<sub>8</sub>, as shown below.

$$4610_8 = 100\ 110\ 001\ 000 = \text{PAGE } 10\ 011 = \text{PAGE } 23_8$$

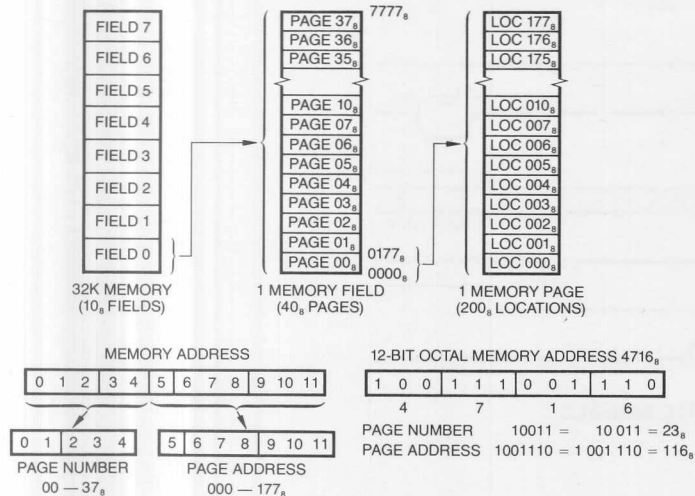
Location 4610<sub>8</sub> is in PAGE 23<sub>8</sub>. Location 123<sub>8</sub> in PAGE 23<sub>8</sub>, CURRENT PAGE, will be:

$$\begin{array}{cc} 10\ 011\ 1\ 010\ 011 & = 100\ 111\ 010\ 011 = 4723_8 \\ \text{PAGE} & \text{PAGE} \\ \text{NUMBER} & \text{ADDRESS} \\ 23_8 & 123_8 \end{array}$$

By this method, 256 locations may be directly addressed, 128 on PAGE 0 and 128 on the CURRENT PAGE. Other locations are addressed by utilizing bit 3. When bit 3 is a 0, the operand address is a DIRECT ADDRESS. An INDIRECT ADDRESS (pointer address) identifies the location that contains the desired address (effective address). To address a location that is not directly addressable, not in PAGE 0 or in the CURRENT PAGE, the absolute address of the desired location is stored in one of the 256 directly addressable locations (pointer address). Upon execution, the MRI will operate on the contents of the location identified by the address contained in the pointer location.

It should be noted that locations 0010<sub>8</sub>-0017<sub>8</sub> in PAGE 0 are AUTOINDEXED. If these locations are addressed indirectly, the contents are incremented by 1 and restored before they are used as the operand address. These locations may, therefore, be used for indexing applications.

Table 1 lists the mnemonics for the five memory reference instruction, their OPCODE, the operations they perform, the number of states and the execution time at +5.0V and +10.0V, assuming a crystal frequency of 3.3MHz, 4MHz and 8MHz or a state time period of 600ns, 500ns and 250ns, respectively.



### MEMORY ORGANIZATION

During an instruction fetch cycle, the IM6100 fetches the instruction pointed to by the PC. The contents of the PC are transferred to the MAR. The PC is incremented by 1. The PC now contains the address of the 'next' sequential instruction. The MAR contains the address of the 'current' instruction which must be fetched from memory. Bits 0-4 of the MAR identify the CURRENT PAGE, that is, the Page from which instructions are currently being fetched and bits 5-11 of the MAR identify the location within the Current Page. (PAGE ZERO (0), by definition, denotes the first 128 words of memory, 0000<sub>8</sub>-0177<sub>8</sub>.)

It should be noted that the data is represented in Two's Complement Integer notation. In this system, the negative of a number is formed by complementing each bit in the data word and adding "1" to the complemented number. The sign is indicated by the most significant bit. In the 12-bit word used by the IM6100, when bit 0 is a "0", it denotes a positive number and when bit 0 is a "1", it denotes a negative number. The maximum number ranges for this system are  $3777_8$  (+2047) and  $4000_8$  (-2048).

Notations applied in Table 1, are defined as follows:

- ( ) Denotes the contents of the register or location within the parenthesis. (EA) is read as "... the contents of the Effective Address".
- (( )) Denotes the contents of the location pointed to by the contents of the location within the double parenthesis. ((PA)) is read as "... the contents of the location pointed to by the contents of the Pointer Address."
- $\leftarrow$  Denotes "... is replaced by ..."
- $\wedge$  Denotes, logical AND operation
- $\vee$  Denotes, logical OR operation

TABLE 1

MNE-MONIC	OP CODE	OPERATION	NUMBER OF STATES	EXECUTION TIME ( $\mu$ s)		
				IM6100 +5.0V 4MHz	IM6100A +10.0V 8MHz	IM6100C +5.0V 3.3MHz
AND	$0_8$	LOGICAL AND DIRECT (I = 0) Operation: $(AC) \leftarrow (AC) \wedge (EA)$ Description: Contents of the EA are logically AND'ed with the contents of the AC and the result is stored in AC.	10	5.0	2.50	6.0
		LOGICAL AND INDIRECT (I = 1, PA $\neq$ 0010-0017 <sub>8</sub> ) Operation: $(AC) \leftarrow (AC) \wedge ((PA))$	15	7.5	3.75	9.0
		LOGICAL AND AUTOINDEX (I = 1, PA = 0010-0017 <sub>8</sub> ) Operation: $(PA) \leftarrow (PA) + 1$ ; $(AC) \leftarrow (AC) \wedge ((PA))$	16	8.0	4.00	9.6
TAD	$1_8$	BINARY ADD DIRECT (I = 0) Operation: $(AC) \leftarrow (AC) + (EA)$ Description: Contents of the EA are ADD'ed with the contents of the AC and the result is stored in the AC; carry out complements the LINK. If AC is initially cleared, this instruction acts as LOAD from Memory	10	5.0	2.50	6.0
		BINARY ADD INDIRECT (I = 1, PA $\neq$ 0010-0017 <sub>8</sub> ) Operation: $(AC) \leftarrow (AC) + ((PA))$	15	7.5	3.75	9.0
		BINARY ADD AUTOINDEX (I = 1, PA = 0010-0017 <sub>8</sub> ) Operation: $(PA) \leftarrow (PA) + 1$ ; $(AC) \leftarrow (AC) + ((PA))$	16	8.0	4.00	9.6
ISZ	$2_8$	INCREMENT AND SKIP IF ZERO DIRECT (I = 0) Operation: $(EA) \leftarrow (EA) + 1$ ; if $(EA) = 0000_8$ , $PC \leftarrow PC + 1$ Description: Contents of the EA are incremented by 1 and restored. If the result is zero, the next sequential instruction is skipped.	16	8.0	4.00	9.6
		INCREMENT AND SKIP IF ZERO INDIRECT (I = 1, PA $\neq$ 0010-0017 <sub>8</sub> ) Operation: $((PA)) \leftarrow ((PA)) + 1$ ; if $((PA)) = 0000_8$ , $PC \leftarrow PC + 1$	21	10.5	5.25	12.6
		INCREMENT AND SKIP IF ZERO AUTOINDEX (I = 1, PA = 0010-0017 <sub>8</sub> ) Operation: $(PA) \leftarrow (PA) + 1$ ; $((PA)) \leftarrow ((PA)) + 1$ ; if $((PA)) = 0000_8$ , $PC \leftarrow PC + 1$	22	11.0	5.50	13.2
DCA	$3_8$	DEPOSIT AND CLEAR THE ACCUMULATOR DIRECT (I = 0) Operation: $(EA) \leftarrow (AC)$ ; $(AC) \leftarrow 0000_8$ Description: The contents of the AC are stored in EA and the AC is cleared.	11	5.5	2.75	6.6
		DEPOSIT AND CLEAR THE ACCUMULATOR INDIRECT (I = 1, PA $\neq$ 0010-0017 <sub>8</sub> ) Operation: $((PA)) \leftarrow (AC)$ ; $(AC) \leftarrow 0000_8$	16	8.0	4.00	9.0
		DEPOSIT AND CLEAR THE ACCUMULATOR AUTOINDEX (I = 1, PA = 0010-0017 <sub>8</sub> ) Operation: $(PA) \leftarrow (PA) + 1$ ; $((PA)) \leftarrow (AC)$ ; $(AC) \leftarrow 0000_8$	17	8.5	4.25	10.2
JMS	$4_8$	JUMP TO SUBROUTINE DIRECT (I = 0) Operation: $(EA) \leftarrow (PC)$ ; $(PC) \leftarrow EA + 1$ Description: The contents of the PC are stored in the EA. The PC is incremented by 1 immediately after every instruction fetch. The contents of the EA now point to the next sequential instruction following the JMS (return address). The next instruction is taken from EA + 1.	11	5.5	2.75	6.6
		JUMP TO SUBROUTINE INDIRECT (I = 1, PA $\neq$ 0010-0017 <sub>8</sub> ) Operation: $((PA)) \leftarrow PC$ ; $(PC) \leftarrow (PA) + 1$	16	8.0	4.00	9.6
		JUMP TO SUBROUTINE AUTOINDEX (I = 1, PA = 0010-0017 <sub>8</sub> ) Operation: $(PA) \leftarrow (PA) + 1$ ; $((PA)) \leftarrow PC$ ; $(PC) \leftarrow (PA) + 1$	17	8.5	4.25	10.2
JMP	$5_8$	JUMP DIRECT (I = 0) Operation: $(PC) \leftarrow EA$ Description: The next instruction is taken from the EA.	10	5.0	2.50	6.0
		JUMP INDIRECT (I = 1, PA $\neq$ 0010-0017 <sub>8</sub> ) Operation: $(PC) \leftarrow (PA)$	15	7.5	3.75	9.0
		JUMP AUTOINDEX (I = 1, PA = 0010-0017 <sub>8</sub> ) Operation: $(PA) \leftarrow (PA) + 1$ ; $(PC) \leftarrow (PA)$	16	8.0	4.00	9.6

MEMORY REFERENCE INSTRUCTIONS

# OPERATE INSTRUCTIONS

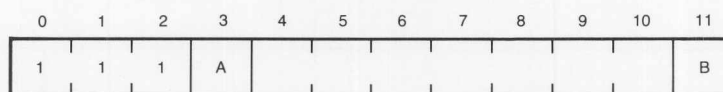
## 10

The Operate Instructions, which have an OPCODE of  $7_8$  (111), consists of 3 groups of microinstructions. Group 1 microinstructions, which are identified by the presence of a 0 in bit 3, are used to perform logical operations on the contents of the accumulator and link. Group 2 microinstructions, which are identified by the presence of a 1 in bit 3 and a 0 in bit 11, are used primarily to test the contents of the accumulator and then conditionally skip the next sequential instruction. Group 3 microinstructions have a 1 in bit 3 and a 1 in bit 11 and are used to perform logical operations on the contents of the AC and MQ.

The basic OPR instruction format is shown in Figure 4.

Operate microinstructions from any group may be microprogrammed with other operate microinstructions of the same group. The actual code for a microprogrammed combination of two, or more, microinstructions is the bitwise logical OR of the octal codes for the individual microinstructions. When more than one operation is microprogrammed into a single instruction, the operations are performed in a prescribed sequence, with logical sequence number 1 microinstructions performed first, logical sequence number 2 microinstructions performed second, logical sequence number 3 microinstructions performed third and so on. Two operations with the same logical sequence number, within a given group of microinstructions, are performed simultaneously.

FIGURE 4



MICROINSTRUCTION	A	B
GROUP 1	0	$\frac{0}{1}$
GROUP 2	1	0
GROUP 3	1	1

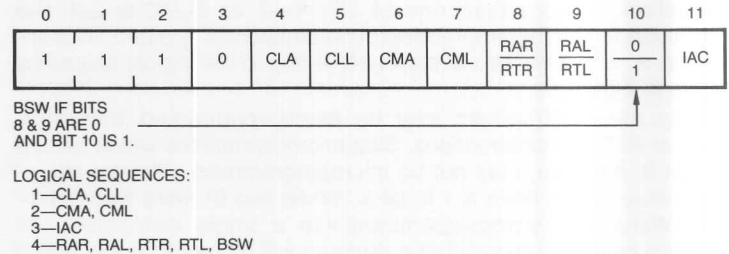
BASIC OPR INSTRUCTION FORMAT

### GROUP 1 MICROINSTRUCTIONS

Figure 5 shows the instruction format of a group 1 microinstruction. Any one of bits 4 to 11 may be set, loaded with a binary 1, to indicate a specific group 1 microinstruction. If more than one of these bits is set, the instruction is a microprogrammed combination of group 1 microinstructions, which will be executed according to the logical sequence shown in Figure 5.

Table 2 lists commonly used group 1 microinstructions, their assigned mnemonics, octal number, instruction format, logical sequence, the operation they perform, the number of states and the execution time at +5.0V and +10.0V, assuming a crystal frequency of 3.3MHz, 4MHz and 8MHz or a state time period of 600ns, 500ns and 250ns, respectively. The same format is followed in Table 3 and 4 which correspond to group 2 and 3 microinstructions, respectively.

### FIGURE 5



### GROUP 1 MICROINSTRUCTION FORMAT

### TABLE 2

MNEMONIC	OCTAL CODE	LOGICAL SEQUENCE	OPERATION	NUMBER OF STATES	EXECUTION TIME ( $\mu$ s)		
					IM6100 +5.0V 4MHz	IM6100A +10.0V 8MHz	IM6100C +5.0V 3.3MHz
NOP	7000	1	NO OPERATION—This instruction causes a 10 state delay in program execution, without affecting the state of the IM6100. It may be used for timing synchronization or as a convenient means of deleting an instruction from a program.	10	5.0	2.50	6.0
IAC	7001	3	INCREMENT ACCUMULATOR—The content of the AC is incremented by one (1) and carry out complements the Link (L).	10	5.0	2.50	6.0
RAL	7004	4	ROTATE ACCUMULATOR LEFT—The contents of the AC and L are rotated one binary position to the left. AC (0) is shifted to L and L is shifted to AC (11).	15	7.5	3.75	9.0
RTL	7006	4	ROTATE TWO LEFT—The contents of the AC and L are rotated two binary positions to the left. AC (1) is shifted to L and L is shifted to AC (10).	15	7.5	3.75	9.0
RAR	7010	4	ROTATE ACCUMULATOR RIGHT—The content of the AC and L are rotated one binary position to the right. AC (11) is shifted to L and L is shifted to AC (0).	15	7.5	3.75	9.0
RTR	7012	4	ROTATE TWO RIGHT—The contents of the AC and L are rotated two binary positions to the right. AC (10) is shifted to L and L is shifted to AC (1).	15	7.5	3.75	9.0
BSW	7002	4	BYTE SWAP—The right six (6) bits of the AC are exchanged or SWAPPED with the left six bits. AC (0) is swapped with AC (6), AC (1) with AC (7), etc. L is not affected.	15	7.5	3.75	9.0
CML	7020	2	COMPLEMENT LINK—The content of the link is complemented.	10	5.0	2.50	6.0
CMA	7040	2	COMPLEMENT ACCUMULATOR—The content of each bit of the AC is complemented having the effect of replacing the content of the AC with its one's complement.	10	5.0	2.50	6.0
CIA	7041	2,3	COMPLEMENT AND INCREMENT ACCUMULATOR—The content of the AC is replaced with its two's complement. Carry out complements the LINK.	10	5.0	2.50	6.0
CLL	7100	1	CLEAR LINK—The link is loaded with a binary 0.	10	5.0	2.50	6.0
CLL RAL	7104	1,4	CLEAR LINK—ROTATE ACCUMULATOR LEFT.	15	7.5	3.75	9.0
CLL RTL	7106	1,4	CLEAR LINK—ROTATE TWO LEFT.	15	7.5	3.75	9.0
CLL RAR	7110	1,4	CLEAR LINK—ROTATE ACCUMULATOR RIGHT.	15	7.5	3.75	9.0
CLL RTR	7112	1,4	CLEAR LINK—ROTATE TWO RIGHT.	15	7.5	3.75	9.0
STL	7120	1,2	SET THE LINK—The LINK is loaded with a binary 1 corresponding with a microprogrammed combination of CLL and CML.	10	5.0	2.50	6.0
CLA	7200	1	CLEAR ACCUMULATOR—The accumulator is loaded with binary 0's.	10	5.0	2.50	6.0
CLA IAC	7201	1,3	CLEAR ACCUMULATOR—INCREMENT ACCUMULATOR.	10	5.0	2.50	6.0
GLT	7204	1,4	GET THE LINK—The AC is cleared; the content of L is shifted into AC (11), a 0 is shifted into L. This is a microprogrammed combination of CLA and RAL.	15	7.5	3.75	9.0
CLA CLL	7300	1	CLEAR ACCUMULATOR—CLEAR LINK.	10	5.0	2.50	6.0
STA	7240	1,2	SET THE ACCUMULATOR—Each bit of the AC is set to 1 corresponding to a microprogrammed combination of CLA and CMA.	10	5.0	2.50	6.0

### GROUP 1 OPERATION MICROINSTRUCTIONS

# OPERATE INSTRUCTIONS CONTINUED

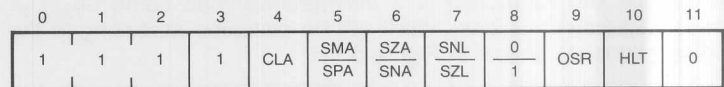
12

## GROUP 2 MICROINSTRUCTIONS

Figure 6 shows the instruction format of group 2 microinstructions. Bits 4—10 may be set to indicate a specific group 2 microinstruction. If more than one of bits 4—7 or 9—10 is set, the instruction is a microprogrammed combination of group 2 microinstructions, which will be executed according to the logical sequence shown in Figure 6.

Skip microinstructions may be microprogrammed with CLA, OSR, or HLT microinstructions. Skip microinstructions which have a 0 in bit 8, however, may not be microprogrammed with skip microinstructions which have a 1 in bit 8. When two or more skip microinstructions are microprogrammed into a single instruction, the resulting condition on which the decision will be based is the logical OR of the individual conditions when bit 8 is 0, or, when bit 8 is 1, the decision will be based on the logical AND.

FIGURE 6



LOGICAL SEQUENCES:  
 1 (Bit 8 is Zero) — SMA or SZA or SNL  
 (Bit 8 is One) — SPA and SNA and SZL  
 2 — CLA  
 3 — OSR, HLT

GROUP 2 MICROINSTRUCTION FORMAT

TABLE 3

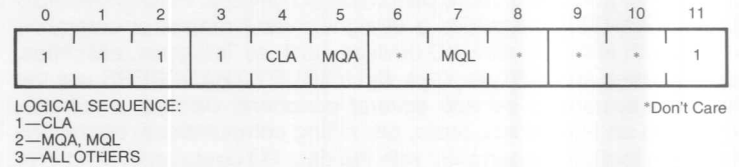
MNEMONIC	OCTAL CODE	LOGICAL SEQUENCE	OPERATION	NUMBER OF STATES	EXECUTION TIME ( $\mu$ s)		
					IM6100 +5.0V 4MHz	IM6100A +10.0V 8MHz	IM6100C +5.0V 3.3MHz
NOP	7400	1	NO OPERATION—See GROUP 1 MICROINSTRUCTIONS	10	5.0	2.50	6.0
HLT	7402	3	HALT—Program stops at the conclusion of the current machine cycle. If HLT is combined with others in OPR 2, the other operations are completed before the end of the cycle.	10	5.0	2.50	6.0
OSR	7404	3	OR WITH SWITCH REGISTER—The content of the Switch Register if OR'ed with the content of the AC and the result is stored in the AC. The OSR INSTRUCTION TIMING is shown in Figure 8. The IM6100 sequences the OSR instruction through a 2-cycle execute phase referred to as OPR 2A and OPR 2B.	15	7.5	3.75	9.0
SKP	7410	1	SKIP—The content of the PC is incremented by 1, to skip the next sequential instruction.	10	5.0	2.50	6.0
SNL	7420	1	SKIP ON NON-ZERO LINK—The content of L is sampled, the next sequential instruction is skipped if L contains a 1. If L contains a 0, the next instruction is executed.	10	5.0	2.50	6.0
SZL	7430	1	SKIP ON ZERO LINK—The content of L is sampled, the next sequential instruction is skipped if L contains a 0. If the L contains a 1, the next instruction is executed.	10	5.0	2.50	6.0
SZA	7440	1	SKIP ON ZERO ACCUMULATOR—The content of the AC is sampled; the next sequential instruction is skipped if the AC has all bits which are 0. If any bit in the AC is a 1, the next instruction is executed.	10	5.0	2.50	6.0
SNA	7450	1	SKIP ON NON-ZERO ACCUMULATOR—The content of the AC is sampled; the next sequential instruction is skipped if the AC has any bits which are not 0. If every bit in the AC is 0, the next instruction is executed.	10	5.0	2.50	6.0
SZA SNL	7460	1	SKIP ON ZERO ACCUMULATOR, <b>OR</b> SKIP ON NON-ZERO LINK, <b>OR</b> BOTH	10	5.0	2.50	6.0
SNA SZL	7470	1	SKIP ON NON-ZERO ACCUMULATOR <b>AND</b> SKIP ON ZERO LINK	10	5.0	2.50	6.0
SMA	7500	1	SKIP ON MINUS ACCUMULATOR—If the content of AC (0) contains a 1, indicating that the AC contains a negative two's complement number, the next sequential instruction is skipped. If AC (0) contains a 0, the next instruction is executed.	10	5.0	2.50	6.0
SPA	7510	1	SKIP ON POSITIVE ACCUMULATOR—The contents of AC (0) are sampled. If AC (0) contains a 0, indicating that the AC contains a positive two's complement number, the next sequential instruction is skipped. If AC (0) contains a 1, the next instruction is executed.	10	5.0	2.50	6.0
SMA SNL	7520	1	SKIP ON MINUS ACCUMULATOR <b>OR</b> SKIP ON NON-ZERO LINK <b>OR</b> BOTH	10	5.0	2.50	6.0
SPA SZL	7530	1	SKIP ON POSITIVE ACCUMULATOR <b>AND</b> SKIP ON ZERO LINK	10	5.0	2.50	6.0
SMA SZA	7540	1	SKIP ON MINUS ACCUMULATOR <b>OR</b> SKIP ON ZERO ACCUMULATOR <b>OR</b> BOTH	10	5.0	2.50	6.0
SPA SNA	7550	1	SKIP ON POSITIVE ACCUMULATOR <b>AND</b> SKIP ON NON-ZERO ACCUMULATOR	10	5.0	2.50	6.0
SMA SZA SNL	7560	1	SKIP ON MINUS ACCUMULATOR <b>OR</b> SKIP ON ZERO ACCUMULATOR <b>OR</b> SKIP ON NON-ZERO LINK <b>OR</b> ALL	10	5.0	2.50	6.0
SPA SNA SZL	7570	1	SKIP ON POSITIVE ACCUMULATOR <b>AND</b> SKIP ON NON-ZERO ACCUMULATOR <b>AND</b> SKIP ON ZERO LINK	10	5.0	2.50	6.0
CLA	7600	2	CLEAR ACCUMULATOR—The AC is loaded with binary 0's.	10	5.0	2.50	6.0
LAS	7604	1,3	LOAD ACCUMULATOR WITH SWITCH REGISTER—The content of the AC is loaded with the content of the SR, bit for bit. This is equivalent to a microprogrammed combination of CLA and OSR.	15	7.5	3.75	9.0
SZA CLA	7640	1,2	SKIP ON ZERO ACCUMULATOR <b>THEN</b> CLEAR ACCUMULATOR	10	5.0	2.50	6.0
SNA CLA	7650	1,2	SKIP ON NON-ZERO ACCUMULATOR <b>THEN</b> CLEAR ACCUMULATOR	10	5.0	2.50	6.0
SMA CLA	7700	1,2	SKIP ON MINUS ACCUMULATOR <b>THEN</b> CLEAR ACCUMULATOR	10	5.0	2.50	6.0
SPA CLA	7710	1,2	SKIP ON POSITIVE ACCUMULATOR <b>THEN</b> CLEAR ACCUMULATOR	10	5.0	2.50	6.0

GROUP 2 OPERATE MICROINSTRUCTIONS

### GROUP 3 MICROINSTRUCTIONS

Figure 7 shows the instruction format of group 3 microinstructions which requires bits 3 and 11 to contain a 1. Bits 4, 5 or 7 may be set to indicate a specific group 3 microinstruction. If more than one of the bits is set, the instruction is a microprogrammed combination of group 3 microinstructions following the logical sequence listed in Figure 7.

### FIGURE 7



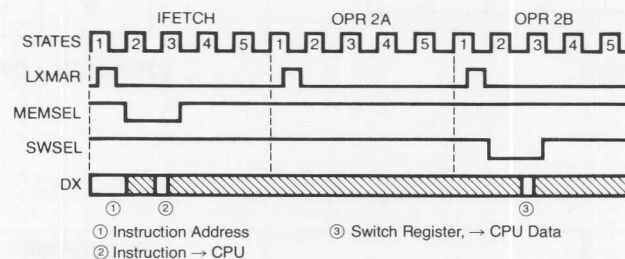
### GROUP 3 MICROINSTRUCTION FORMAT

### TABLE 4

MNEMONIC	OCTAL CODE	LOGICAL SEQUENCE	OPERATION	NUMBER OF STATES	EXECUTION TIME ( $\mu$ s)		
					IM6100 +5.0V 4.0MHz	IM6100A +10.0V 8.0MHz	IM6100C +5.0V 3.3MHz
NOP	7401	3	NO OPERATION—See Group 1 Microinstructions	10	5.0	2.50	6.0
MQL	7421	2	MQ REGISTER LOAD—The content of the AC is loaded into the MQ, the AC is cleared and the original content of the MQ is lost.	10	5.0	2.50	6.0
MQA	7501	2	MQ REGISTER INTO ACCUMULATOR—The content of the MQ is OR'ed with the content of the AC and the result is loaded into the AC. The original content of the AC is lost but the original content of the MQ is retained. This instruction provides the programmer with an inclusive OR operation.	10	5.0	2.50	6.0
SWP	7521	3	SWAP ACCUMULATOR AND MQ REGISTER—The content of the AC and MQ are interchanged accomplishing a microprogrammed combination of MQA and MQL.	10	5.0	2.50	6.0
CLA	7601	1	CLEAR ACCUMULATOR	10	5.0	2.50	6.0
CAM	7621	3	CLEAR ACCUMULATOR AND MQ REGISTER—The content of the AC and MQ are loaded with binary 0's. This is equivalent to a microprogrammed combination of CLA and MQL.	10	5.0	2.50	6.0
ACL	7701	3	CLEAR ACCUMULATOR AND LOAD MQ REGISTER INTO ACCUMULATOR—This is equivalent to a microprogrammed combination of CLA and MQA.	10	5.0	2.50	6.0
CLA SWP	7721	3	CLEAR ACCUMULATOR AND SWAP ACCUMULATOR AND MQ REGISTER—The content of the AC is cleared. The content of the MQ is loaded into the AC and the MQ is cleared.	10	5.0	2.50	6.0

### GROUP 3 OPERATE MICROINSTRUCTIONS

### FIGURE 8



### OSR INSTRUCTION TIMING

The input/output transfer instructions, which have an OPCODE of 6<sub>8</sub>, are used to initiate the operation of peripheral devices and to transfer data between peripherals and the IM6100. Three types of data transfer may be used to receive or transmit information between the IM6100 and one or more peripheral I/O devices. PROGRAMMED DATA TRANSFER provides a straightforward means of communicating with relatively slow I/O devices, such as Teletypes, cassettes, card readers and CRT displays. INTERRUPT TRANSFERS use the interrupt system to service several peripheral devices simultaneously, on an intermittent basis, permitting computational operations to be performed concurrently with the data I/O operations. Both Programmed Data Transfers and Program Interrupt Transfers use the accumulator as a buffer, or storage area, for all data transfers. Since data may be transferred only between the accumulator and the peripheral, only one 12 bit word at a time may be transferred. DIRECT MEMORY ACCESS, DMA, transfers variable-size blocks of data between high-speed peripherals and the memory with a minimum of program control required by the IM6100.

### IOT INSTRUCTION FORMAT

The Input/Output Transfer Instruction format, the number of states and the execution time at +5.0V and +10.0V, assuming a crystal frequency of 3.3MHz, 4MHz and 8MHz or a state time period of 600ns, 500ns and 250ns, respectively is represented in Figure 9.

The first three bits, 0-2, are always set to 6<sub>8</sub> (110) to specify an IOT instruction. The next six bits, 3-8, contain the device selection code that determines the specific I/O device for which the IOT instruction is intended and, therefore, permit interface with up to 64 I/O devices. The last three bits, 9-11, contain the operation specification code that determines the specific operation to be performed. The nature of this operation for any given IOT instruction depends entirely upon the circuitry designed into the I/O device interface.

### PROGRAMMED DATA TRANSFER

Programmed Data Transfer is the easiest, simplest, most convenient and most common means of performing data I/O. For micro-processor applications, it may also be the most cost effective

approach. The data transfer begins when the IM6100 fetches an instruction from the memory and recognizes that the current instruction is an IOT. This is referred to as IFETCH and consists of five (5) internal states. The IM6100 sequences the IOT instruction through a 2-cycle execute phase referred to as IOT<sub>A</sub> and IOT<sub>B</sub>. Bits 0-11 of the IOT instruction are available on DX 0-11 at IOT<sub>A</sub> · LXMAR. These bits must be latched in an external address register. DEVSEL is active low to enable data transfers between the IM6100 and the peripheral device(s). Input-Output Instruction Timing is shown in Figure 10. The selected peripheral device communicates with the IM6100 through 4 control lines—C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub> and SKP. In the IM6100 the type of data transfer, during an IOT instruction, is specified by the peripheral device(s) by asserting the control lines as shown in Table 5.

The control line SKP, when low during an IOT, causes the IM6100 to skip the next sequential instruction. This feature is used to sense the status of various signals in the device interface. The C<sub>0</sub>, C<sub>1</sub>, and C<sub>2</sub> lines are treated independently of the SKP line. In the case of a RELATIVE or ABSOLUTE JUMP, the skip operation is performed after the jump. The input signals to the IM6100, DX 0-11, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, and SKP, are sampled at IOT<sub>A</sub> during DEVSEL · XT<sub>C</sub>. The data from the IM6100 is available to the device(s) during DEVSEL · XT<sub>C</sub>. IOT<sub>B</sub> cycle is internal to the IM6100 to perform the operations requested during IOT<sub>A</sub>. Both IOT<sub>A</sub> and IOT<sub>B</sub> consist of six (6) internal states.

In summary, Programmed Data Transfer performs data I/O with a minimum of hardware support. The maximum rate at which programmed data transfers may take place is limited by the IM6100 instruction execution rate. However, the data rate of the most commonly used peripheral devices is much lower than the maximum rate at which programmed transfers can take place in the IM6100. The major drawback associated with Programmed Data Transfer is that the IM6100 must hang up in a waiting loop while the I/O device completes the last transfer and prepares for the next transfer. On the other hand, this technique permits easy hardware implementation and simple, economical interface design. For this reason, almost all devices except bulk storage units rely heavily on programmed data transfer for routine data I/O.

FIGURE 9

0	1	2	3	4	5	6	7	8	9	10	11
1	1	0									

IOT INSTRUCTION FORMAT

NUMBER OF STATES	EXECUTION TIME (μs)		
	IM6100 +5.0V 4MHz	IM6100A +10.0V 8MHz	IM6100C +5.0V 3.3MHz
17	8.5	3.4	10.2

IOT NUMBER OF STATES/EXECUTION TIME

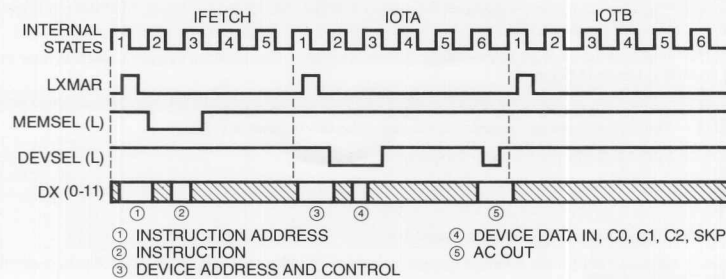
TABLE 5

CONTROL LINES C <sub>0</sub> C <sub>1</sub> C <sub>2</sub>			OPERATION	DESCRIPTION
H	H	H	DEV ← AC	The content of the AC is sent to the device.
L	H	H	DEV ← AC; CLA	The content of the AC is sent to a device and then the AC is cleared.
H	L	H	AC ← AC V DEV	Data is received from a device, OR'ed with the data in the AC and the result is stored in the AC.
L	L	H	AC ← DEV	Data is received from a device and loaded into the AC.
*	H	L	PC ← PC + DEV	Data from the device is added to the contents of the PC. This is referred to as a RELATIVE JUMP.
*	L	L	PC ← DEV	Data is received from a device and loaded into the PC. This is referred to as an ABSOLUTE JUMP.

\*Don't Care

PROGRAMMED I/O CONTROL LINES

FIGURE 10



INPUT-OUTPUT INSTRUCTION TIMING

## INTERRUPT TRANSFER

### PROGRAM INTERRUPT TRANSFERS

The program interrupt system may be used to initiate programmed data transfers in such a way that the time spent waiting for device status is greatly reduced or eliminated altogether. It also provides a means of performing concurrent programmed data transfers between the IM6100 and the peripheral devices. This is accomplished by isolating the I/O handling routines from the mainline program and using the interrupt system to ensure that these routines are entered only when an I/O device status is set, indicating that the device is actually ready to perform the next data transfer, or that it requires some sort of intervention from the running program.

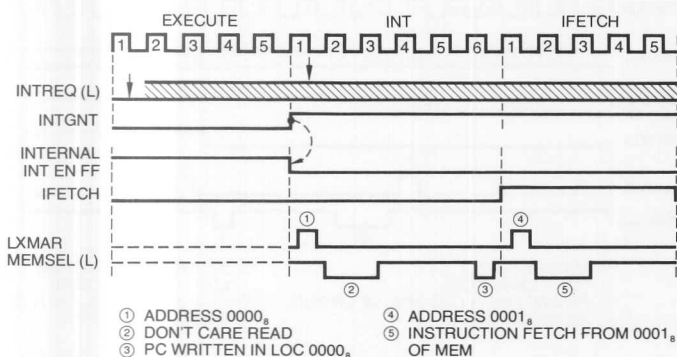
The interrupt system allows certain external conditions to interrupt the computer program by driving the INTREQ input to the IM6100 Low. If no higher priority requests are outstanding and the interrupt system is enabled, the IM6100 grants the device interrupt at the end of the current instruction. After an interrupt has been granted, the Interrupt Enable Flip-Flop in the IM6100 is reset so that no more interrupts are acknowledged until the interrupt system is re-enabled under program control.

### DEVICE INTERRUPT GRANT TIMING

The current content of the Program Counter, PC, is deposited in location 0000<sub>8</sub> of the memory and the program fetches the instruction from location 0001<sub>8</sub>. The return address is available in location 0000<sub>8</sub>. This address must be saved in a software stack if nested interrupts are permitted. The INTGNT, Figure 11, signal is activated by the IM6100 when a device interrupt is acknowledged. This signal is reset by executing any IOT instruction as shown in Figure 12. The INTGNT signal is necessary to implement the Extended Memory Control hardware when more than 4K of memory is required. The INTGNT is also useful in implementing an External Vectored Priority Interrupt network.

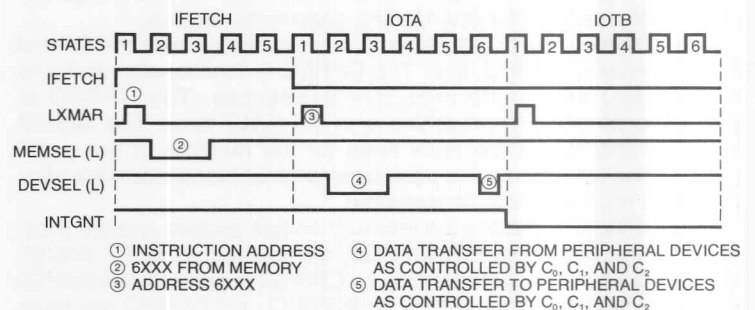
The user program controls the interrupt mechanism of the IM6100 by executing the processor IOT instructions listed in Table 6. Several of these interrupt IOT instructions are also used if the memory is extended beyond 4K words.

FIGURE 11



DEVICE INTERRUPT GRANT TIMING

FIGURE 12



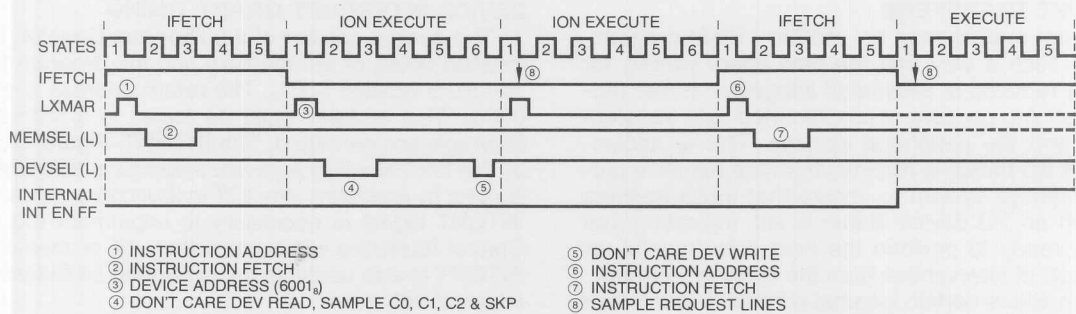
DEVICE INTERRUPT GRANT RESET TIMING

TABLE 6

MNE-MONIC	OCTAL CODE	OPERATION
SKON	6000	SKIP IF INTERRUPT ON—If Interrupt system is enabled, the next sequential instruction is skipped. The Interrupt system is disabled.
ION	6001	INTERRUPT TURN ON—The internal interrupt acknowledge system is enabled. The interrupt system is enabled after the CPU executes the next sequential instruction. The INTERRUPT ENABLE TIMING is shown in Figure 13.
IOF	6002	INTERRUPT TURN OFF—The interrupt system is disabled. Note that the interrupt system is automatically disabled when the CPU acknowledges an INT request.
SRQ	6003	SKIP IF INT REQUEST—The next sequential instruction is skipped if the INT request bus is low.
GTF	6004	GET FLAGS—The following machine states are read into the indicated bits of AC. bit 0—Link bit 2—INT request bus bit 4—Interrupt Enable FF  Other bits may be modified by external inputs (ex. Extended memory control).
RTF	6005	RETURN FLAGS—Link is restored from AC (0). Interrupt system is enabled after the next sequential instruction is executed. All AC bits are available externally to restore external states. (ex. Extended memory control).
SGT	6006	Operation is determined by external devices, if any.
CAF	6007	CLEAR ALL FLAGS—AC and Link are cleared. Interrupt system is disabled.

PROCESSOR IOT INSTRUCTIONS

FIGURE 13



INTERRUPT ENABLE FF ON (ION)

CONTROL PANEL INTERRUPT TRANSFER

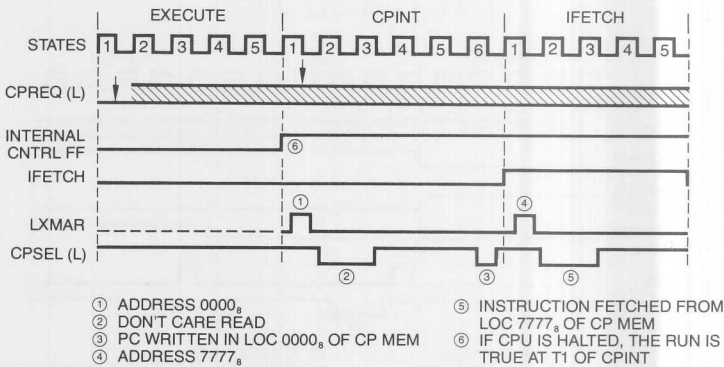
The IM6100 control panel is implemented in software. The software implementation of the control panel need not use any part of the main memory or change the processor state. This is an important feature since the final version of the system may not have a control panel and the system designer would like to use the entire capacity of the main memory for the specific system application.

The control panel communicates with the IM6100 with the Control Panel Request, CPREQ, line. The CPREQ is functionally similar to the INTREQ with some important differences. The CPREQ is granted even when the machine is in the HALT state. The IM6100 is temporarily put in the RUN state for the duration of the panel routine. The IM6100 reverts back to its original processor state after the panel routine has been executed.

The CPREQ bypasses the interrupt enable system and the processor IOT instructions, ION and IOF, are ignored while the IM6100 is in the Control Panel Mode. Once a CPREQ is granted, the IM6100 will not recognize any DMAREQ or INTREQ until CPREQ has been fully serviced.

When a CPREQ is granted, Figure 14, the PC is stored in location 0000<sub>8</sub> of the Panel Memory and the IM6100 resumes operation at location 7777<sub>8</sub> of the Panel Memory. The Panel Memory would be organized with RAM's in the lower pages and PROM's in the higher pages. The control panel service routine would be stored in the higher pages in the nonvolatile PROM's, starting at 7777<sub>8</sub>.

FIGURE 14

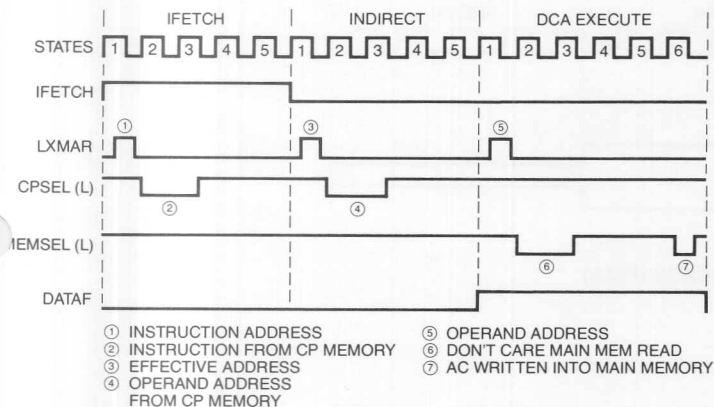


CONTROL PANEL INTERRUPT GRANT TIMING

A Control Panel Flip-Flop, CNTRL FF, which is internal to the IM6100, is set when the CPREQ is granted. The CNTRL FF prevents further CPREQ's from being granted.

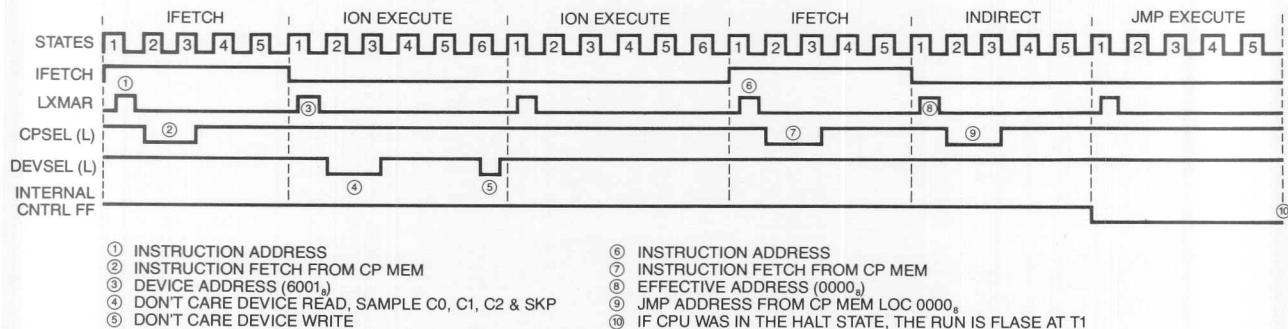
As long as the CNTRL FF is set, the Control Panel Memory Select, CPSEL, is active instead of the Memory Select, MEMSEL, for memory references. The CPSEL signal may, therefore, be used to distinguish the Control Panel Memory from the Main Memory. However, during the Execute phase of indirectly addressed AND, TAD, ISZ or DCA instructions, the MEMSEL is made active. The instructions are always fetched from the control panel memory. The operand address for indirectly addressed AND, TAD, ISZ or DCA refers first to the control panel memory for an effective address, which, in turn, refers to a location in the main memory. A main memory location may, therefore, be examined and changed by indirectly addressed TAD and DCA instructions, Figure 15, respectively. Every location in the main memory is accessible to the control panel routine.

FIGURE 15



"DCA INDIRECT" IN CONTROL PANEL ROUTINE

FIGURE 16



"ION; JMP I 0000" IN CONTROL PANEL ROUTINE

Exiting from the control panel routine is achieved by executing the following sequence with reference made to Figure 16.

ION

JMP I 0000<sub>8</sub> (Loc 0000<sub>8</sub> in CPMEM)

The ION, 6001<sub>8</sub>, instruction will reset the CP FF after executing the next sequential instruction. The ION will not affect the interrupt system since the CNTRL FF is still active. Location 0000<sub>8</sub> of the CPMEM contains either the original return address deposited by the IM6100 when the CP routine was entered, or it may be a new starting address defined by the CP routine, for example, by activating the LOAD ADDRESS SWITCH. CPREQ's are normally generated by the manual actuation of the control switches. If the CPU registers must be displayed in real-time, the CPREQ's must be generated by a timer at fixed intervals.

The designer may also make use of the control panel features to implement Bootstrap loaders in the CP Memory so that the loader will be "transparent" to the main memory. Programs will be loaded by DCA I POINTER instruction, the pointer being developed in the CP RAM to point to the main memory location to be loaded.

Approximately 64 P/ROM locations are sufficient to implement all the functions of the PDP8/E Control Panel. The IM6100 provides for a 12 bit switch register which can be read by the IM6100 under program control with the OR THE SWITCH REGISTER, OSR, instruction even without a control panel.

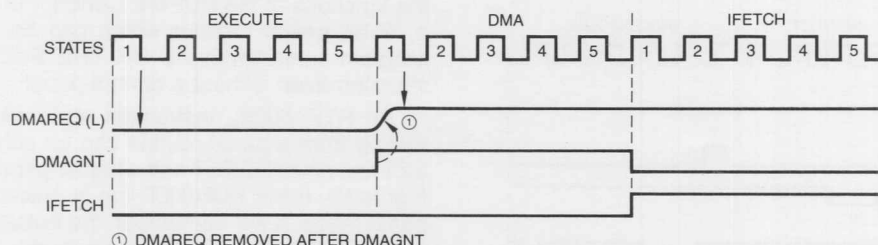
An RTF, 6005<sub>8</sub>, instruction also resets the internal CNTRL FF. Exiting from a panel routine can be achieved by activating the RESET line since RESET has a higher priority than CPREQ as shown in Figure 18. If the RUN/HLT line is pulsed while the IM6100 is in the panel mode, it will 'remember' the pulse(s) but defer any action until the IM6100 exits from the panel mode.

### DIRECT MEMORY ACCESS (DMA)

Direct Memory Access, sometimes called data break, is the preferred form of data transfer for use with high-speed storage devices such as magnetic disk or tape units. The DMA mechanism transfers data directly between memory and peripheral devices. The IM6100 is involved only in setting up the transfer; the transfers take place with no processor intervention on a "cycle stealing" basis. The DMA transfer rate is limited only by the bandwidth of the memory and the data transfer characteristics of the device.

The device generates a DMA Request when it is ready to transfer data. The IM6100 grants the DMAREQ by activating the DMAGNT signal at the end of the current instruction as shown in Figure 17. The IM6100 suspends any further instruction fetches until the DMAREQ line is released. The DX lines are tri-stated, all SEL lines are high, and the external timing signals  $XT_A$ ,  $XT_B$ , and  $XT_C$  are active. The device which generated the DMAREQ must provide the address and the necessary control signals to the memory for data transfers. The DMAREQ line can also be used as a level sensitive "pause" line.

FIGURE 17



① DMAREQ REMOVED AFTER DMAGNT

### DIRECT MEMORY ACCESS (DMA)

# INTERNAL PRIORITY STRUCTURE

19

After an instruction is completely sequenced, the major state generator scans the internal priority network as shown in Figure 18. The state of the priority network decides the next sequence of the IM6100.

The request lines, RESET, CPREQ, RUN/HLT, DMAREQ and INTREQ, are sampled in the last cycle of an instruction execution, at time T<sub>1</sub>. The worst case response time of the IM6100 to an external request is, therefore, the time required to execute the longest instruction preceded by any 6-state execution cycle. For the IM6100, this is an autoindexed ISZ, 22 states, preceded by any 6-state execution cycle instruction. The worst case response time is, therefore, 28 states, 14  $\mu$ s at 5 volts.

When the IM6100 is initially powered up, the state of the timing generator is undefined. The generator is automatically initialized with a maximum of 68 clock pulses. The request inputs, as the IM6100 is powered on, must span at least 92 clock pulses to be recognized, 68 clocks for the counter to initialize and a maximum of two IM6100 cycles (20 to 24 clocks) for the state generator to sample the request lines.

The internal priority is RESET, CPREQ, RUN/HLT, DMAREQ, INTREQ, and IFETCH.

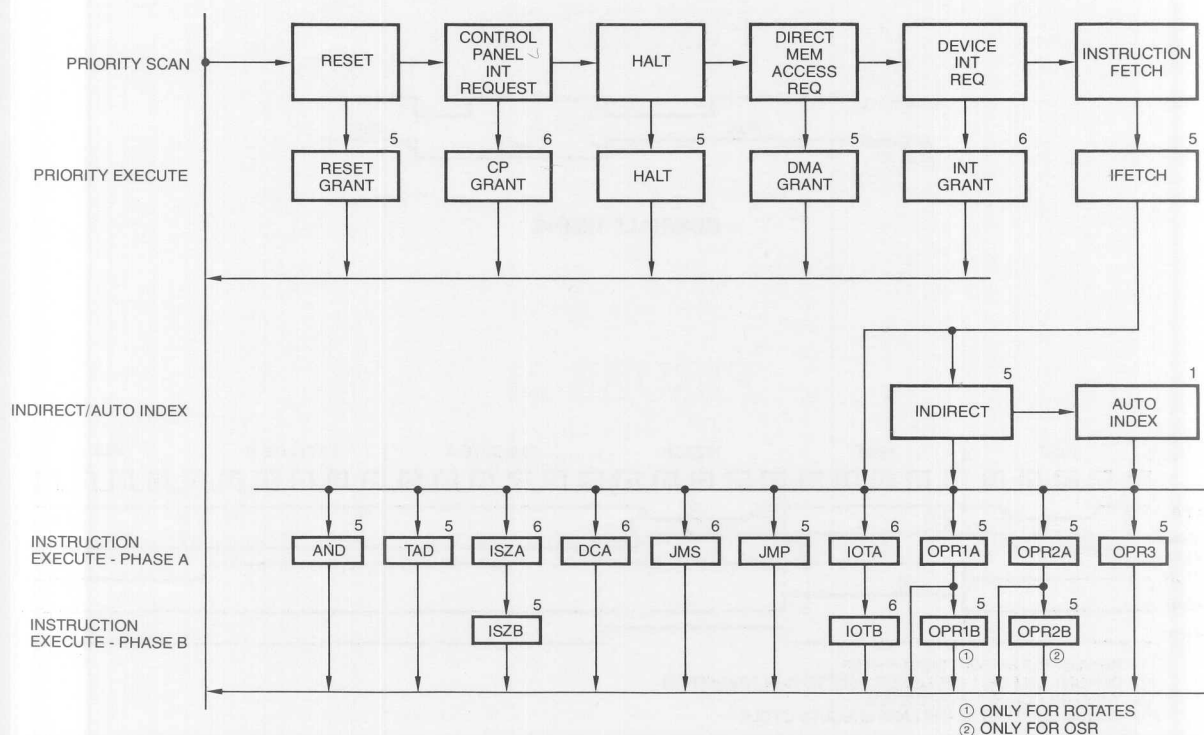
## IFETCH

If no external requests are pending, the IM6100 fetches the next instruction pointed to by the contents of the PC. The IFETCH line is active during the cycle in which the instruction is fetched. External devices can monitor DX, 0-2, during IFETCH·XT<sub>A</sub> to determine the functional class of the current instruction. For example, the external memory extension hardware must know when JMP or JMS instructions are fetched to implement the Extended Memory Control.

The Programmable Logic Array, PLA, in the IM6100 sequences the IM6100 to execute the fetched instruction. All INDIRECT and AUTOINDEX Memory Reference Instructions go through a common state sequence to generate the Effective Address, EA, of the operand. The subsequent sequence, referred to as the EXECUTE phase, is controlled by the functional class of the instruction. The EXECUTE phase of AND, TAD, DCA, JMS, JMP and OPR Group 3 Microinstruction consists of only one cycle. ISZ and IOT have a 2-cycle EXECUTE phase. OPR Group 1 and Group 2 Microinstructions have an optional second cycle, depending on the microcoding of the OPR instructions. An IM6100 cycle consists of 5 states, T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub>, T<sub>4</sub>, and T<sub>5</sub>, with an optional sixth state, T<sub>6</sub>, for Output Transfers (WRITE).

The state sequence for internal (processor) and external IOT instructions are identical. The Device Address and Control bits are available in the External Address Register for internal IOT instructions. External hardware, for example Extended Memory Control, can control the C-lines for data transfers to implement Get Flags (GTF), Return Flags (RTF), and Clear All Flags (CAF) instructions. External Control of the C-lines is necessary to implement these internal IOT instructions since the flag bits may be distributed both inside and outside the IM6100.

FIGURE 18



MAJOR PROCESSOR STATES AND NUMBER OF CLOCK CYCLES IN EACH STATE

# INTERNAL PRIORITY STRUCTURE CONTINUED

20

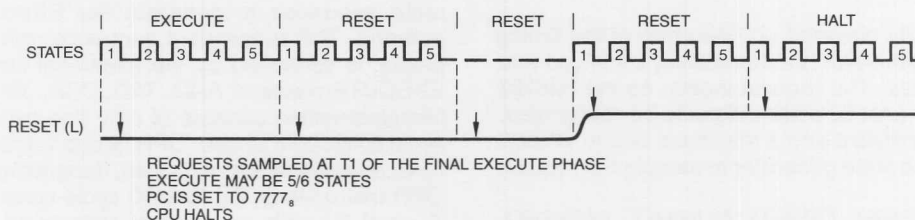
## RESET

The Reset initializes all internal IM6100 flags and clears the AC and the LINK. The machine is halted.

The IM6100 remains in the Reset state as long as the Reset line is low as shown in Figure 19. The DX lines are three stated. The IM6100 continues to provide the external timing signals XT<sub>A</sub>, XT<sub>B</sub> and XT<sub>C</sub>. All SEL lines are high.

The PC is set to 7777<sub>h</sub>. In most applications, the higher memory locations utilize P/ROM's or ROM's. Therefore, a power-up routine starting at the highest memory location can be used to initialize the system.

FIGURE 19



RESET TIMING

## RUN/HALT

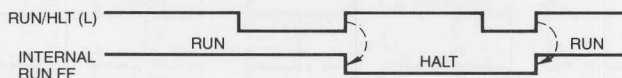
RUN/HLT changes the state of the IM6100's RUN/HLT flip-flop. Pulsing the line low causes the IM6100 to alternately run and halt as shown in Figure 20. The RUN/HLT line is normally high. The IM6100 recognizes the positive transition of the signal.

The RUN/HLT flip-flop can be put in the halt state under program control by executing the HLT, 7402<sub>h</sub>, instruction. When the IM6100 is halted, RUN/HLT is functionally identical to the CONTINUE switch of the PDP8/E control panel.

If the IM6100 is in the halt state, the RUN signal is low. The RUN signal can be used to power down external circuitry for a low power system.

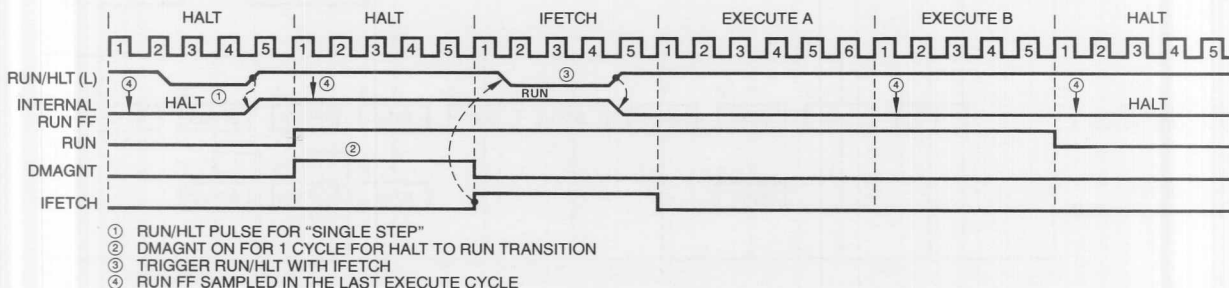
The RUN/HLT can also be used to make the IM6100 execute one instruction at a time as shown in Figure 21. The RUN/HLT combines the functional features of STOP, CONTINUE, and SINGLE INSTRUCTION as defined by the PDP8/E Control Panel.

FIGURE 20



RUN/HALT TIMING

FIGURE 21



"SINGLE STEP" WITH RUN/HLT

The IM6100 and the PDP-8/E\* of Digital Equipment Corporation are software compatible. The basic PDP-8/E paper-tape software system supplied by Digital Equipment Corporation will operate properly with the IM6100. This basic software package includes Binary Loaders, PAL III Assembler, Symbolic Editor, Dynamic Debugging Technique (DDT), Octal Debugging Technique (ODT), 23 Bit Floating Point Package and FORMula CALCulator (FOCAL)\*. The IM6100 will execute the complete set of CPU diagnostics for PDP-8/E.

Since the bus structure of the IM6100 can be adapted to provide a subset of the PDP-8/E OMNIBUS\* signals, as shown in Figure 22, all programmed I/O interfaces for the PDP-8/E, for example, Teletype, Papertape Reader/Punch, etc., will operate with the IM6100 without any hardware or software modification.

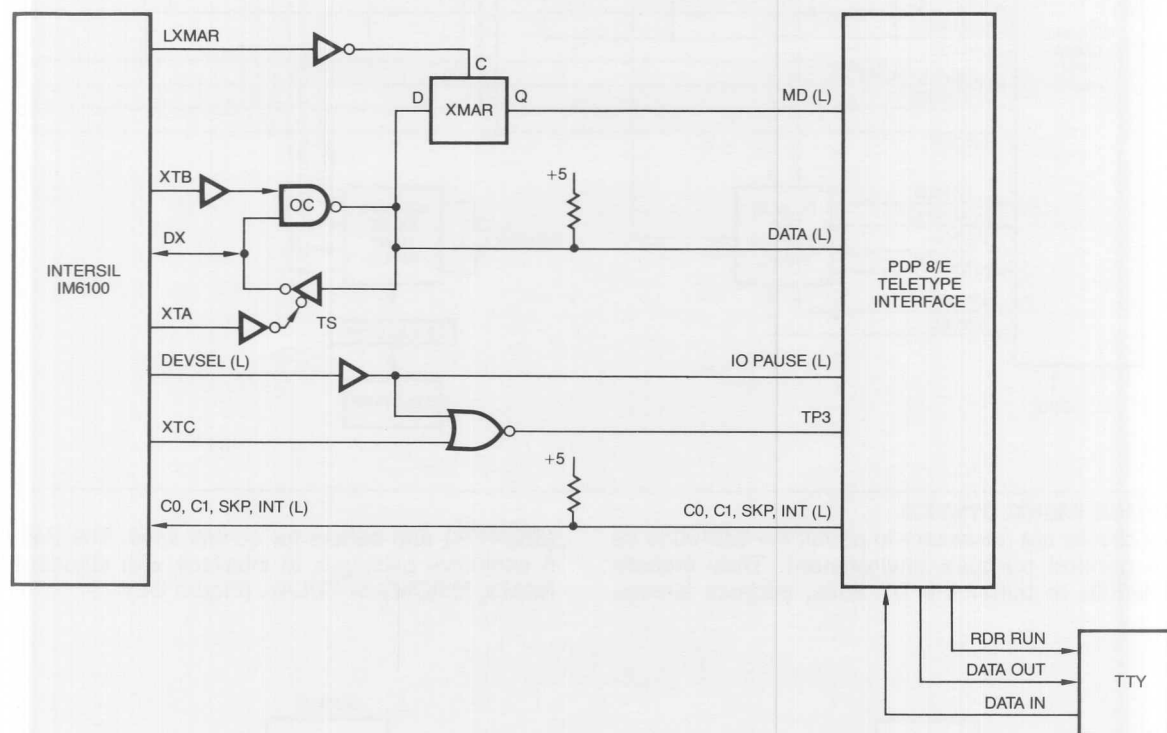
The Direct Memory Access, DMA, structure of the IM6100 and PDP-8/E are different, the IM6100 DMA structure is similar to the PDP-8 1-CYCLE BREAK, but not compatible.

The IM6100 handles 4K words of memory directly. Like the PDP-8/E, an external Extended Memory Control element can be used to extend the addressing space up to 32K. All necessary control and timing signals to implement the memory extension controller are generated by the IM6100.

The Extended Arithmetic Element, EAE, and the User Flag, UF, processor options of the PDP-8/E cannot be used with the IM6100. The EAE is used for hardwired Multiply/Divide and the UF for timesharing.

The IM6100 treats the Control Panel as a programmed I/O device with certain special features. The Control Panel has a dedicated INT request line to the IM6100 and the control panel program can reside in a separate memory, distinct from the normal program memory. The control panel service routine can be made transparent to the user and the user program can occupy the entire 4K of main memory. The bootstrap routines may also reside in the dedicated control panel memory. Unlike the PDP-8/E, the IM6100 bootstrap routines and the loaded user programs, can, therefore, share common address space.

FIGURE 22



EXAMPLE OF A PDP-8/E PROGRAMMED I/O PERIPHERAL INTERFACE

# APPLICATIONS

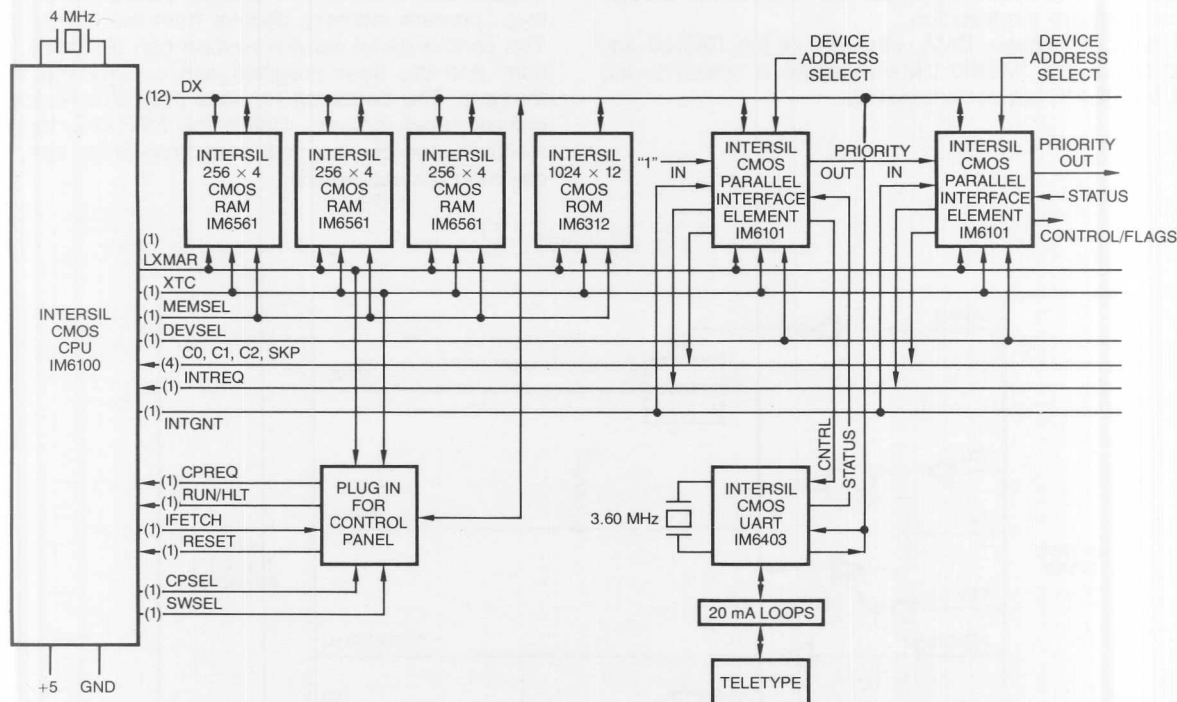
22

## ALL CMOS SYSTEM

The IM6100 microprocessor family provides for the capability of building an all CMOS system with no additional support components. The CMOS RAM devices are organized  $256 \times 1$  (IM6524),  $1024 \times 1$ , (IM6508/18) or  $256 \times 4$  (IM6551/6561). They have internal address latches and operate synchronously with an address strobe. A  $1024 \times 12$  bit mask programmable CMOS ROM (IM6312) is also provided. The IM6402/6403 is an industry standard UART with the option of

operating directly from a high frequency crystal. The IM6101, Parallel Interface Element (PIE), provides all the signals necessary to communicate with an external device including a vectored priority interrupt chain. For example, a parallel Teletype interface can be designed with only two logic elements—the IM6101 for control and the IM6403 for data handling. The dynamic power dissipation of the CMOS system will be less than 60mW at +5 volts. (Figure 23.)

FIGURE 23

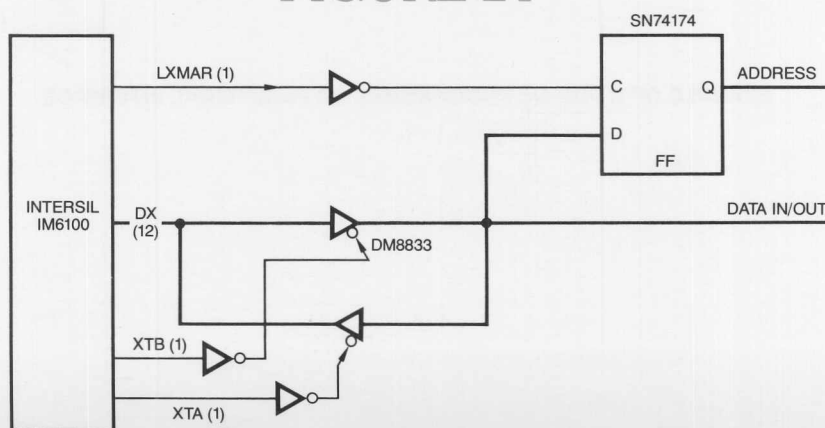


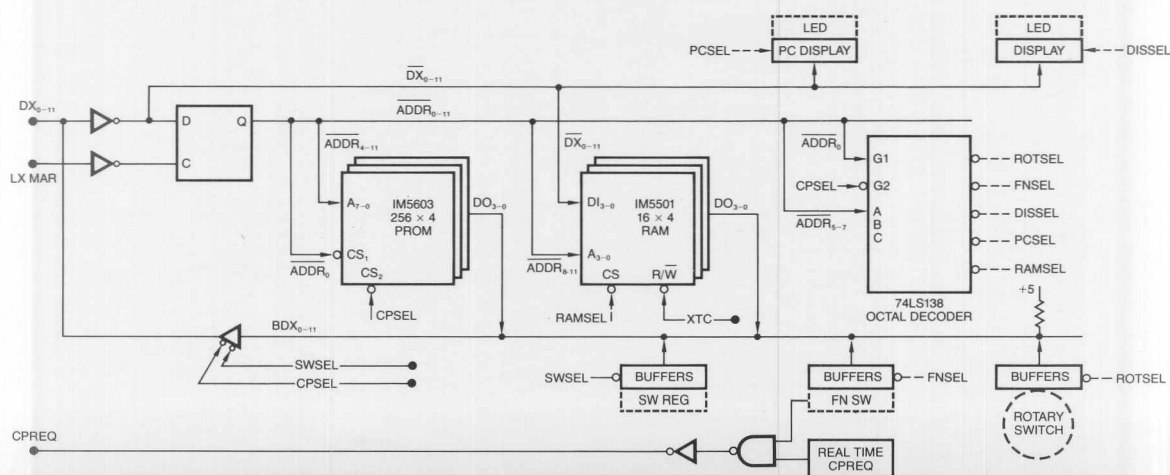
## GENERAL PURPOSE IM6100 SYSTEM

A few auxiliary circuits are necessary to permit the IM6100 to be operational in a general purpose environment. They include transceivers (DM8833) to buffer the DX lines, address latches

(SN74174) and buffers for control lines. The IM6100 requires only 6 additional packages to interface with standard bipolar or MOS RAM's, P/ROM's or FPLA's. (Figure 24.)

FIGURE 24





**IM6100 TO CMOS RAM INTERFACE**

The IM6100 provides all the control signals to interface directly with standard CMOS RAM's. Since the CMOS RAM's have internal address latches, the address information on the DX lines is latched

internally with the address strobe. Address, Data-in and Data-out can be multiplexed on the DX lines without any degradation in performance. (Figure 27.)

FIGURE 27

